Principles of Cryptography Problem Sheet 5

Dr. Kurunandan Jain[1]†,

[1]Department of Mathematics, Amrita Vishwa, Amrita Campus, Kollam-690525, Kerala, India

(Received xx; revised xx; accepted xx)

1) If a message from Alice to Bob is found to be authentic, i.e. iin fact originated from Alice, integrity is automatically assured, since an alteration by Oscar would make him the originator. However, this can't be the case if sender authenticity is assured.

b) No, a message can still be unaltered but message authenticity is not given. For instance, Oscar could masquerade as Alice and send a message to Bob saying that is it from Alice. Although the message arrives unaltered at Bob's (integrity is thus assured) send authenticity is given.

2) Threats: Unauthorized physical access to the building, to the data, to internal network

Social engineering

Data might be modified, secret data might be read

Key generation might be predictable or weak

Key encryption might be weak

Integrity measures used might be weak

Data might be lost

Trust by the users might be lost

Organization and physical measures:

Physical access control to the building

Training and guidelines for the personnel

Secure backup procedures

IT security functions:

Key generation (random number generator of good quality)

Key distribution (encryption+ securing for data integrity of the session key)

Access control of the network

Timestampe service

3) This is a valid signature, one must check $x \equiv 6292^{131} \mod 9797$

ii) Invalid signature

iii) Valid signature since $x \equiv 1424^{131} \mod 9797$

4) Oscar receives the message, alters it and signs it with his own private key a'. Then he sends the new message together with the signature and the public key (n', e') of Alice (which is isntead the one of Oscar)

5) In order for a signature to be verified $\beta^r r^s \equiv \alpha^x \mod p$

From the question we know that $\alpha^x = 3^{10} \equiv 25 \mod 31$

Now, $t = \beta^r r^s \equiv p \equiv 6^{17} \times 17^5 \mod 31 = 25$

Since $\alpha^x \mod p$ is also 25, we arrive at the signature is valid.

ii)$t = \beta^r r^s = 6^{13} \times 13^{15} \equiv 6 \times 30 \equiv 25 \mod 31$ which again gives a valid signature

b) Due to the fact that Elgamal signature scheme is probabilistic, there are $p - 1$, i.e. in this case 30 different signatures for each message x

6)

7)

† Email address for correspondence: kurunandanj@am.amrita.edu

$$
\begin{aligned}
s_1 &\equiv (x_1 - dr)\, k_{E_1}^{-1} \bmod p-1 \\
s_2 &\equiv (x_2 - dr)\, k_{E_2}^{-1} = (x_2 - dr)\,(k_{E_1}+1)^{-1} \bmod p-1 \\
\Rightarrow \frac{s_1}{s_2} &\equiv \frac{(x_1 - dr)\,(k_{E_1}+1)}{(x_2 - dr)\, k_{E_1}} \bmod p-1 \\
\Leftrightarrow k_{E_1} &= \frac{1}{\frac{s_1 (x_2 - dr)}{s_2 (x_1 - dr)} - 1} \bmod p-1 \\
\Rightarrow d &\equiv \frac{x_1 - s_1 k_{E_1}}{r} \bmod p-1
\end{aligned}
$$

FIGURE 1. Question 6 solution

10.15 Similarly to the attack on Elgamal, an attacker can use following system of equations

$$
\begin{aligned}
s_1 &\equiv (SHA(x_1) + dr)\, k_E^{-1} \bmod q \\
s_2 &\equiv (SHA(x_2) + dr)\, k_E^{-1} \bmod q
\end{aligned}
$$

for known s_1, s_2, x_1, and x_2 to first compute the ephemeral key k_E and then the private key d:

$$
\begin{aligned}
s_1 - s_2 &\equiv k_E^{-1}(SHA(x_1) - SHA(x_2)) \bmod q \\
\Leftrightarrow k_E &\equiv \frac{SHA(x_1) - SHA(x_2)}{s_1 - s_2} \bmod q \\
\Rightarrow d &\equiv \frac{s_1 \cdot k_E - SHA(x_1)}{r} \bmod q
\end{aligned}
$$

FIGURE 2. Question 7 solution

1. $A_1 = E_0 + f_1(B_0, C_0, D_0) + (A)_{<<<5} + W_j + K_t = 5A827999_{hex}\, B_1 = A_0 = 00000000_{hex}\, C_1 = (B_0)_{<<<30} = 00000000_{hex}\, D_1 = C_0 = 00000000_{hex}\, E_1 = D_0 = 00000000_{hex}$
2. $A_1 = E_0 + f_1(B_0, C_0, D_0) + (A)_{<<<5} + W_j + K_t = 6A827999_{hex}\, B_1 = A_0 = 00000000_{hex}\, C_1 = (B_0)_{<<<30} = 00000000_{hex}\, D_1 = C_0 = 00000000_{hex}\, E_1 = D_0 = 00000000_{hex}$

FIGURE 3. Question 9 solution

8) $t = 2^{\frac{65}{2}}\sqrt{\ln\frac{1}{1-0.5}}$ $t = 2^{\frac{129}{2}}\sqrt{\ln\frac{1}{1-0.5}}$ $t = 2^{\frac{161}{2}}\sqrt{\ln\frac{1}{1-0.5}}$
$t = 2^{\frac{65}{2}}\sqrt{\ln\frac{1}{1-0.1}}$ $t = 2^{\frac{129}{2}}\sqrt{\ln\frac{1}{1-0.1}}$ $t = 2^{\frac{161}{2}}\sqrt{\ln\frac{1}{1-0.1}}$

9)

10) Calculate $x||h = e_{k_1}^{-1}(y)$

Calculate $h' = H(k_2||x)$

If $h' = H$ the message is authentic. If $h \neq h'$ either the message of the MAC (or both) has been altered during transfer

Protocol B:

Calculate $x||s = e_{k_1}^{-1}(y)$

10) As we have seen MACs can be used to authentic messages. With this problem, we want to show the difference between two protocols one with a MAC, one with a digital signature. In the two protocols, the sending party performs the following operation

Protocol A: $y = e_{k_1}[x||h(k_2||x]$

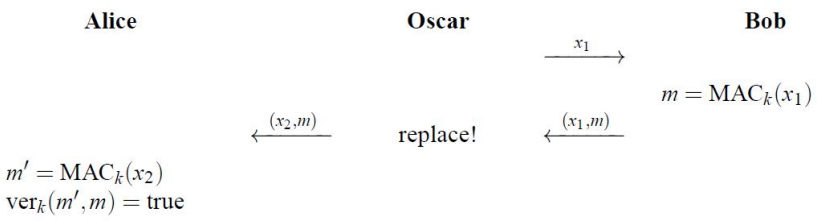

FIGURE 4. Question 11 solution

where x is the message, $h()$ is a hash function such as SHA-1, e is a private-key encryption algorithm, $||$ denotes simple concatenation and k_1, k_2 are secret keys which are only known to send and the receiver.

Protocol B: $y = e_k[x||sig_{k_{pr}}(h(x))]$

Calculate $h' = H(x)$

Verify the signature $ver_{k_{pub}}(s, H(x))$

11) This attack assumes that Oscar can trick Bob into signing the message x_1. This is, of course, not possible in every situation, but one can imagine scenarios where Oscar can pose as an innocent party and x_1 is the message being generated by Oscar. See Figure 4

ii) For constructing collisions, Oscar must be able to compute about $\sqrt{2^n}$ MACs where, n is the output width. Since Oscar does not have the secret key, he has to somehow trick Alice and/or Bob into computing MACs for that many messages. This is in practice often impossible.

On the other hand, collisions for hash functions can be constructed by Oscar by himself without the help of Alice and Bob because these computations are un-keyed.

An 80 bit MAC provides thus a security of 2^{80} since collisions attacks are not applicable. A hash function with the same output size offers only a security of about 2^{40}

12)

i) Session keys are derived by a linear and invertible operation of the previous session key

ii) Usage of hash functions, thus a non-linear correlation of the session keys

iii) Usage of masterkey and the previous session key for every derivation of the next session key

b) Methods ii and iii provide Perfect Forward Secrecy, since the old session keys cannot be extracted from the recent session key

c) For case i) every session since PFS is missing. For case ii) Every session using the hacked session key K_n and every following session

iii) Only the recent session, since the unknown masterkey is used for every key derivation

d) None, since all the sessions keys can be calculated

13) Once Alice's KEK k_A is being compromised, Oscar can compute the session key k_{ses} and thus decrypt all the messages

b) The same applies to a compromised KEK k_B of Bob.

14) Alice: $A = 2^{228} \equiv 394 \mod 467$

$k_{AO} = O^a = 156^{228} \equiv 243 \mod 467$

Bob: $B = 2^{57} \equiv 313 \mod 467$

$k_{BO} = O^b = 156^{57} \equiv 438 \mod 467$

Oscar: $O = 2^{16} \equiv 156 \mod 467$

$k_{AO} = A^o = 394^{16} \equiv 243 \mod 467$

$k_{BO} = B^o = 313^{16} \equiv 438 \mod 467$